

12



AD A127323

Report No. 5188

Research in Knowledge Representation For Natural Language Understanding

Annual Report
1 September 1981 to 31 August 1982

Prepared for:
Defense Advanced Research Projects Agency

DTIC
ELECTED
S APR 28 1983
A

DTIC FILE COPY

This document has been approved
for public release and sale; its
distribution is unlimited.

88 04 27 003

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BBN Report No. 5188	2. GOVT ACCESSION NO. AD-A127323	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RESEARCH IN KNOWLEDGE REPRESENTATION FOR NATURAL LANGUAGE UNDERSTANDING - Annual Report 1 September 1981 - 31 August 1982		5. TYPE OF REPORT & PERIOD COVERED Annual Report 9/1/81 - 8/31/82
7. AUTHOR(s) Woods, W., Bates, M., Bobrow, R., Goodman, B., Israel, D., Schmolze, J., Schudy, R., Sidner, C., Vilain, M.		6. PERFORMING ORG. REPORT NUMBER BBN Report No. 5188
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 10 Moulton Street Cambridge, MA 02238		8. CONTRACT OR GRANT NUMBER(s) N00014-77-C-0378
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, VA 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE September 1982
		13. NUMBER OF PAGES 53
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Knowledge representation, natural language understanding, artificial intelligence, cognitive science, semantic networks, KL-ONE, parsing, AT's grammars, syntax, semantics, pragmatics, speaker meaning, RUS, natural language interfaces, parallel algorithms, decision aiding, decision support.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) BBN's ARPA project in Knowledge Representation for Natural Language Under- standing is aimed at developing techniques for computer assistance to a decision maker in understanding a complex system or situation using natural language control of an intelligent graphics display. The work that we have been doing falls into three classes: fluent natural language under- standing in a graphics context - including helpful systems that go beyond mere passive execution of literal instructions, fundamental problems of cont'd		

20. Abstract (cont'd)

knowledge representation and use, and abstract parallel algorithms for knowledge base inferential operations. *This report gives* ~~In this report, we will give~~ a brief summary of the activities of this research project during the past year. In addition, ~~we document reports, publications, and presentations~~ during the past year, *are documented.*

Report No. 5188

**RESEARCH IN KNOWLEDGE REPRESENTATION
FOR NATURAL LANGUAGE UNDERSTANDING**

Annual Report
1 September 1981 to 31 August 1982

Principal Investigator:
William A. Woods
(617) 497-3361

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

ARPA Order No. 3414

Contract No. N00014-77-C-0378

Effective Date of Contract:
1 September 1977

Contract Expiration Date:
30 September 1984

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-77-C-0378. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

Accession For	
NTIS G-AT	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
Availability Codes	
Dist	Special



TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
2. RESEARCH ON THE KL-ONE SYSTEM	5
2.1 Summary of the KL-ONE Language	5
2.2 KL-ONE Workshop	5
2.3 Assertions in KL-ONE	6
3. RESEARCH ON PARALLEL ALGORITHMS	9
4. RESEARCH ON NATURAL LANGUAGE UNDERSTANDING	11
4.1 Experimental Determination of the Requisites of Natural Language Interfaces	11
4.2 Lexicon and Coverage of RUS Parser	12
4.3 Changes to the RUS Parser Control Structure	13

5.	THE RUS PARSER CONTROL STRUCTURE	15
5.1	Data Structures Used by RUS	16
5.2	The Parsing Process	21
5.2.1	Processing Push Arcs	21
5.2.2	Processing VIR Arcs	22
5.2.3	Processing POP Arcs	23
5.3	Conclusion	24
6.	TEMPORAL REASONING	25
6.1	By Means of Introduction	25
6.2	A Logic of Time	26
6.3	Using the Logic	30
6.4	Truth Maintenance	32
6.5	Time Points	34
6.6	Absolute Dating	37
6.7	Other Aspects of the System	38
7.	PUBLICATIONS	41
8.	BIBLIOGRAPHY AND REFERENCES	49

LIST OF FIGURES

FIG. 1.	PRIMITIVE RELATIONS BETWEEN INTERVALS	27
FIG. 2.	THE RELATION A (DURING BEGINS OVERLAPS) B	28
FIG. 3.	ILLUSTRATIONS OF RULES 2-4	29
FIG. 4.	CONSTRAINT PROPAGATION	31
FIG. 5.	NEW PRIMITIVE RELATIONS (INVOLVING POINTS)	35
FIG. 6.	ILLUSTRATIONS OF RULES 5-7	36

1. INTRODUCTION

BBN's ARPA project in Knowledge Representation for Natural Language Understanding is aimed at developing techniques for computer assistance to a decision maker in understanding a complex system or situation. The motivating need is that of a commander in a command and control context both in strategic situation assessment and in more tactical situations. The emphasis is on the performance of the decision maker and his computer systems in crisis situations, where the commander needs an extremely flexible system which is capable of manipulating large amounts of data and presenting it in a variety of ways until the commander feels satisfied that he has a grasp of the situation.

Such a system would have the ability to present tabular, graphical, cartographic, and other information in ways which are comprehensible and appropriate to the situation. It would have the ability to display many different map overlays and the ability to change the display to meet changing needs. Such a system must also have the ability construct and use models of the domain and dialog and it should also have the ability to organize and present the information at various levels of detail, including knowledge and data supporting any reduction or summarizing processes. Techniques to produce such displays on demand, in response to high level descriptions of what they should contain, do not currently exist, and will require breakthroughs in areas of language understanding, knowledge representation, and knowledge based inference.

The work which we have been doing falls into three classes, successively motivated by the initial goal of providing powerful

computer assistance to a commander in a complex decision-making task. One such class is fluent natural language understanding. Robustness, fluency, flexibility, and ease of use requires systems which go beyond mere passive execution of literal instructions. Abilities are required to represent and use models of the strategic and tactical domains, of the discourse between the system and the commander, of the intentions of the commander, and of the pragmatics of the command and control situation. Thus our second area of work is the representation of the many kinds of knowledge which are essential to the performance of such a system. A third area of our work has been directed at the design of advanced parallel algorithms and systems which can support the decision maker in real time.

A major accomplishment in this work so far has been the development of the knowledge representation system KL-ONE. KL-ONE has been adopted by a number of groups throughout the U.S., where it has been used for applications ranging from natural language understanding to VLSI design. KL-ONE has now been transported to a variety of computer architectures and languages; versions have now been implemented in SmallTalk for various Xerox processors, and in FranzLISP for the Digital Equipment VAX series and others, as well as a number of machines (including the PDP-20, BBN Jericho and the Xerox Dolphin) which support InterLISP.

A component of this project is devoted to cooperation with the ARPA sponsored Consul group at ISI, to provide them with current versions of the RUS parsing system and the KL-ONE knowledge representation system, and to work with them on knowledge representation problems which arise out of that work. This cooperation has continued to be very fruitful for both

groups. Major products of this cooperation during this year have included significant contributions to our understanding of the knowledge components and features of KL-ONE which are most useful in actual system implementations, and the knowledge constructs which are less frequently used and can be naturally and faithfully expressed using other KL-ONE language constructs without degrading the expressive power or ease of use of KL-ONE. This body of understanding is expected to play a major role next year in a redesign and reimplementaion of KL-ONE which should be both more expressive and more efficient.

Major KL-ONE activities this year have included the 1981 KL-ONE Workshop, which gathered researchers from twenty-one universities and research institutions to discuss KL-ONE, and a conceptual design of extensions of KL-ONE to implement assertions in an incrementally efficient fashion, including detection and support of inconsistencies, the retraction of assertions, and the maintenance of support structures and inferences between assertions. These areas are summarized in chapter two of this report, and detailed in the technical reports and publications listed and described at the end of this report.

Major natural language activities last year which will continue into the next include the design and implementation of a new control structure for the RUS parser, which has significantly improved its efficiency and real time response characteristics, the development of a representation and system for reasoning about points and intervals of time, and a linguistic analysis of the lexicon and coverage of the RUS parser. These activities are summarized in chapters four, five, and six of this report.

An additional natural language research activity this last

year has been in the structure and representation of natural language discourse and the focussing processes in discourse which underly deixis and the resolution of anaphora. Experiments have been performed to determine the necessary characteristics of a combined natural language and graphics interface; in these experiments subjects communicated via a combination of a computer terminal system and an optical graphics system with a pointing device. Command and control, information exchange and other dialogs were explored in this experimental setting; these studies contributed significantly to our understanding of the human communication which must be supported in a natural language and graphics interface. This research is summarized in the natural language component of this report, and detailed in publications summarized at the end of this report.

2. RESEARCH ON THE KL-ONE SYSTEM

2.1 Summary of the KL-ONE Language

The status of the KL-ONE language is summarized on pages 233 through 260 of in the 1981 KL-ONE Workshop Proceedings, which is BBN Report No. 4842. The KL-ONE summary describes in a tutorial style the structure and philosophy of the language, providing a conceptual summary of the types of KL-ONE objects and their structural relations to one another. The terminology and graphic forms of KL-ONE and the JARGON interface to KL-ONE are described and then used in a progression of examples of increasing complexity illustrating the representational issues of KL-ONE.

2.2 KL-ONE Workshop

The Second KL-ONE Workshop gathered researchers from twenty one universities and research institutions for a series of discussions and presentations about the KL-ONE knowledge representation language. This year we opted for a two-part Workshop, the first comprising three days of intensive technical discussions by a small group (14 participants) intimately involved with KL-ONE development, the second comprising two days of presentations and small group discussions.

The technical discussions that preceded the main conference covered areas of current central concern to KL-ONE and knowledge representation in general, including "realization" (attributing new descriptions to individuals as they are learned about) and

"classification" (putting KL-ONE descriptions into a taxonomy according to their internal structure); Individual Concepts (the way to represent definite descriptions in KL-ONE); "Role Set Relations" (the way to represent constraints in concept definitions in KL-ONE) and "Qua-Concepts" (concepts defined as functions of other concepts); and some system maintenance and utility issues (KL-ONE is implemented in INTERLISP at BBN and Smalltalk at Xerox PARC). To allow us to get right to work, the chairman of each session circulated a position paper to the group in advance, raising the questions he wanted to see addressed at the Workshop.

For the general conference session (attended by 46 people), we invited groups from various sites to report on interesting applications of KL-ONE, problems with it, interesting technical questions, etc. Topics of the talks included "KloneTalk" (the version of KL-ONE implemented in SmallTalk - this included a videotaped demonstration of the system's interface), prototypes in knowledge representation, translation of INTERLISP KL-ONE to FranzLisp, a calculus of Structural Descriptions, and the KL-ONE Classifier, not to mention several others. We also had the larger group break up into smaller working groups to consider inference in KL-ONE, representing beliefs, some KL-ONE practice examples, and transporting KL-ONE to other machines. All of these topics are covered in the Workshop Proceedings, which are published as BBN Report No. 4842.

2.3 Assertions in KL-ONE

KL-ONE currently has an exceptionally good representation

for the inheritance relations among structured concepts, including the relations between the corresponding parts of their structures. However, there are many subtleties of representation that are still undergoing active investigation as part of the knowledge representation effort and that require continued development. One major KL-ONE activity this year has centered around the representation of assertions in KL-ONE. The major result of this effort this year has been a conceptual design of an assertion system for KL-ONE. The results of this design effort are expected to include increased efficiency in future systems as well as a conceptual unification and strengthening of many of the KL-ONE features, such as nexuses, in which assertional concepts are currently expressed.

3. RESEARCH ON PARALLEL ALGORITHMS

During this last year we have continued our work on parallel marker passing, and on languages and systems for parallel algorithms and the description of parallel architectures. Two generations of simulators for marker passing machines have been constructed for use as experimental vehicles to advance our understanding of marker passing algorithms.

We have also begun investigating the problems of highly parallel machines such as the connection machine under development in the AI laboratories at MIT. One aspect of this research has been an exploration of the design for a programming system for a broad class of different parallel architectures. One component of such a system is a language in which algorithms can be expressed in such a way that they can be translated into code for a variety of parallel machines. Another component is a language to describe the architectures of parallel machines in such a way and in sufficient detail that a description of a target machine could be interpreted by a language translator, which could then produce a program for that target machine. We have explored the design of languages for both of these aspects of parallel programming, by developing portions of a parallel machine architecture specification language, and of a machine independent parallel programming language. This work is still in progress.

4. RESEARCH ON NATURAL LANGUAGE UNDERSTANDING

Research on Natural Language understanding this year has centered on the structure and representation of dialogs involving natural language and graphics, the coverage of English by the RUS system, and the control structure of the RUS parser.

4.1 Experimental Determination of the Requisites of Natural Language Interfaces

Many research and applications groups are attempting to develop natural language interfaces to systems of many different types. The domain being used, the degree of "intelligence" the system should exhibit, and other characteristics greatly affect the way the language capability should be designed. However there is no generally accepted (or even commonly used) method of determining, in the early stages of the design process, just what capacities the particular natural language interface must possess in order to be effective. In a paper presented to the ECICS-82 [Bates and Sidner82], we set forth a case study of a methodology that has been extremely effective in our domain and which can easily be adapted to other situations.

The particular task we explored was that of a decision maker examining and modifying a database using a graphics display. The decision maker is expected to manipulate both the content of the database and the form of the display. The system is expected to be a helpful, intelligent assistant with considerable linguistic capability so that the user can express commands, questions, facts, and other material very naturally.

To help us understand the special issues of language processing in this environment, we collected protocols of users interacting with simulated versions of the system we envision. Our analysis of those protocols convinced us that we needed a system with very different kinds of linguistic capabilities than are required in environments without graphics or with restrictions on the kind of utterances the user may produce.

The paper presents three aspects of our research on a system that can provide graphically represented information and can talk naturally with a user about that information:

1. description of the methodology that we used in developing and analyzing an extended prototypical dialogue between a user and such a system,
2. portions of our analysis of that dialogue that present both the information obtained and the method of obtaining it, and
3. conclusions about the necessary linguistic and non-linguistic capacities of an intelligent conversational partner, as drawn from the full analysis.

4.2 Lexicon and Coverage of RUS Parser

In order to determine the coverage of the RUS parser, we are attempting to compare it with the variety of English grammars developed in the linguistic community. We want to provide a concise description of the coverage and lexicon of RUS which will be accessible to a broader community. To this end, we are preparing a catalog of English syntactic forms, their descriptions in the various grammars of English, and their representation and coverage in RUS. The study, which has thus

far included the syntactic forms of verbs and their qualifiers, has helped us improve the structure and generality of the RUS system, principally by identifying generalizations of the RUS grammar which include less frequently occurring English constructs.

4.3 Changes to the RUS Parser Control Structure

In addition to improvements in the coverage of the RUS grammar, we have continued to develop and improve the RUS parsing system. In particular, the control structure of the parser has been considerably enhanced, resulting in greatly improved performance. Chapter 5 of this report describes the control structure and serves to document important modifications at both the theoretical and practical levels.

5. THE RUS PARSER CONTROL STRUCTURE

R. Bobrow and M. Bates

Work on the RUS system this year has produced theoretical and practical improvements to the efficiency and flexibility of the system. These advances have included improvements to the RUS grammar and extensions to the control and data structures which make the parsing nearly deterministic. There were three primary techniques used to enhance the performance of the RUS parser:

1. pruning incorrect parse paths
2. eliminating redundancy
3. reordering alternatives

Substantial pruning has been achieved by grammar changes (e.g., lookahead at PUSHes and other critical parts of the grammar), GROUP arcs, and the semantic interface. There is some question about whether it is better to perform lookahead tests on PUSHes (and not set up a generator for the constituent in the well-formed-substring-table (WFST)) or to permit the lower process to begin and almost immediately fail (recording this fact for other processes to notice); work is continuing to evaluate these alternatives.

Redundancy in path-following through the grammar has been virtually eliminated by the use of trace equations in structure representation (see Section 5.2.2) and some changes in the WFST (notably checking for equal structures created on different paths and checking for use of the HOLD list).

Reordering alternatives has been made not only possible but

easy by having multiple agendas and the ReSchedule action available in the grammar. This makes it possible to experiment with depth-, breadth-, and best-first strategies and comparative scheduling processes for activities such as selective modifier placement. All of these mechanisms are discussed in some detail below.

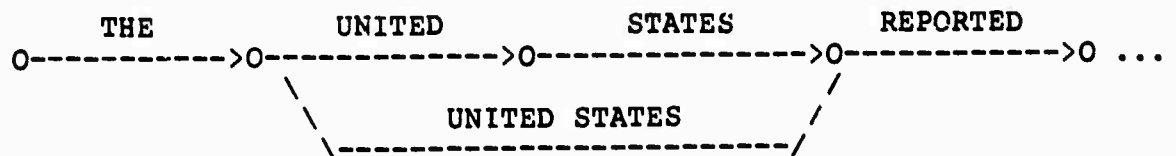
5.1 Data Structures Used by RUS

There are four primary data structures that are used in the RUS parser: machines, configurations, agendas, and machine invocations. Many instances of these structures are created during the parsing process. In addition, there are two important global structures: the chart and the well-formed-substring table (WFST).

The chart is a graph whose edges represent the possible sequences of lexical items in the utterance being analyzed. For a very simple sentence, the chart may not branch at all:

the new idea
0----->0----->0----->0 ...

When there are alternative lexical items, the chart may branch:



When complete constituents are found, they are added to the chart. Thus at any node in the chart, one can determine the next possible word strings and the constituents that have been found to begin at that point.

The WFST can be thought of as a set of buckets each of which contains information about actual and potential syntactic constituents. Each bucket is indexed by the chart node at which the constituent starts, the start state of the level of the grammar which will parse the constituent, and the initial registers (usually NIL) that provide the context for the constituent. Each bucket contains consumers (machine invocations ready to pick up a completed constituent and continue parsing with it) and productions (actual constituents that have been found, together with some associated information). The basic data structures can be schematically characterized as follows:

WFST = set of BUCKETS

BUCKET = CONSUMERS
 + PRODUCTIONS
 + PRODUCER (i.e., a machine)
 + SUSPENDED VIR ARCS

CONSUMER = MACHINE INVOCATION (slightly modified to allow for a constituent to be inserted when it is found)

PRODUCTION = STRUCTJRE OF CONSTITUENT
 + CHART NODE WHERE CONSTITUENT ENDS
 + REGS TO BE RAISED WITH CONSTITUENT (usually NIL)
 + TRACES (filled and unfilled)

A machine corresponds intuitively to one level of a non-deterministic ATN grammar which is looking for a constituent of a particular type at a particular place in the input. (Thus we sometimes talk about an NP machine, or two PP machines which look for prepositional phrases at different points in a sentence.)

MACHINE = CHART POSITION OF FIRST WORD OF CONSTITUENT
 + START STATE OF THIS GRAMMAR LEVEL (this specifies
 both the type of constituent and the function
 which parses it)
 + INITIAL REGISTER LIST (usually NIL)
 + STACK
 + AGENDA

Note that the components of a machine overlap the specification of a bucket in the WFST. The effect is that since buckets are unique, a machine implicitly specifies the place in the WFST where its result (if it succeeds in producing a constituent) will go.

An agenda is a set of configurations in which the machine can be (re)started. It can be considered to be a black box out of which comes the next configuration for the machine to enter. It is actually an ordered list of lists, each of which can be treated as a stack, a queue, or a set of configurations ordered by some weight or score.

AGENDA = ordered set of Qs

Q = NAME(s)
 + ordered set of CONFIGURATIONS

Each machine has its own agenda which tells it what to do next. If the agenda is completely empty, then either the machine has never been run or it has been run to exhaustion.

In addition to these local agendas, the Parser function maintains a global agenda of machine invocations (instead of configurations). When a machine is running, it can create configurations and place them on its own agenda; it can also create machine invocations and place them on the global agenda.

The names of the Qs on the global agenda and their ordering in the current RUS system are as follows:

Priority Q Name(s)

- 1 RESULTS (used for top level POPs only, i.e., success)
- 2 CONSUMERS, PRODUCERS (PUSHes and POPs are high priority)
- 3 MAINSCHEDULE (for consumers consuming an item)
- 4 SUSPENDS, CONDITIONALVIRARCQUEUE (for VIR arcs that are
running because a context now has a usable HOLD list)
VERBFINALNPWITHPPMOD, UNLIKELYSMOD
- 5 REMAININGMACHINES (alternatives to a PUSH arc)
- 6 POPALTS, WAITS, UNLIKELYPRENOUNHEADALTS
(from local machines that succeed;
these are invocations that would restart lower levels)
- 7 PARTITIVEELLIPSISWAITS (e.g., the third of x)
- 8 LASTMODIFIERINTHEREINSERTION, PASSIVEHAVEVERBMOD,
UNLIKELYWHIZCLAUSE

5.2 The Parsing Process

When a machine is (re)started in a particular configuration, it examines the arcs remaining in the current state for one that can be taken. If no such arc is found, then the machine removes the current configuration from its agenda and starts up the next configuration at the top of its agenda. If it finds an arc that can be taken, it creates a configuration embodying the alternative arcs and puts this configuration in its agenda (to be retrieved in case of backup); then it processes the actions on the arc, updating the variables that hold the components of the current active configuration. For certain kinds of arcs, notably PUSH, POP and VIR arcs, the processing is more complex.

5.2.1 Processing Push Arcs

When a PUSH arc is encountered, the machine which is executing suspends itself and allows control to be taken over by the global agenda, but first it makes changes in the agendas and the WFST. If there is no bucket in the WFST at the current position for the kind of constituent the PUSH will look for, then such a bucket is created, a producer machine is created and put in the bucket, a machine invocation for that machine is created and placed on the global agenda, a consumer machine is created (to process the constituent when and if it is found) and is placed in the bucket, and a configuration embodying the alternative arcs to the PUSH arc is created and scheduled in the global agenda (on the REMAININGMACHINES queue).

If there is already a bucket in the WFST for the kind of constituent the PUSH will look for, then a new consumer is

created and attached to the bucket. A configuration embodying the alternatives to the PUSH arc is created and scheduled as above. If there is already one or more constituents (productions) in the bucket, then machine invocations embodying the consumption of those constituents are created and placed in the global agenda.

5.2.2 Processing VIR Arcs

In the original ATN parser, items could be placed on the hold list by one level of the grammar to be picked up by a lower level. This meant that if two parse paths reached the same PUSH arc with different hold lists, two different producers had to be constructed and run. The mechanism of the hold list has been modified to avoid this inefficiency. Now, only one producer is constructed, and when it encounters a VIR arc its action depends upon whether or not there is a consumer at any level above the current one that has an unmatched HOLD list item. If so, the machine creates a trace, which can be thought of as a hole in a constituent. (The terminology comes from linguistics, where a part of a constituent can be moved outside the constituent boundaries by certain transformations, but a "trace" of the item is left behind.) When the constituent has been completed, it is entered in the WFST and can be used by any consumer that has an appropriate item on its hold list. An equation is established between the trace and the real item that corresponds to it.

If, on the other hand, no likely consumer was on the stack when the VIR arc was encountered, then a configuration embodying the VIR arc is saved in a list associated with the WFST bucket. At any time in the future, if a consumer which has a HOLD list is

attached to that bucket (or if such a consumer is created at any level above that bucket!), the VIR arc configuration will be restarted. This is a rather complex mechanism, but it ensures that VIR arcs are taken if and only if there is a HOLD list around that might provide the necessary constituent.

5.2.3 Processing POP Arcs

One check that must be made when an apparently successful POP is encountered is to see whether any items that were placed on the HOLD list at this level are still there. Since items must be used at or below the level where they are held, failure to consume a HOLD list item must result in failure of the POP arc. Another useful check is to see whether the structure being returned from the POP is identical to any of the productions already in the WFST at this point. If so, it means that the same structure has been created by two different paths; this may reflect an error in the grammar, but in any case it would be redundant to include the new constituent in the WFST, so the POP arc is aborted.

If these two tests are passed, then the machine finds all the consumers in the WFST that are waiting for this constituent,¹ creates consuming configurations for them and places those machine invocations in the global agenda on the CONSUMERS queue.

¹Note that if the constituent has a trace in it, i.e., if a VIR arc was taken during its construction, then only the consumers that have appropriate HOLD lists are considered.

5.3 Conclusion

The preceding description gives a rough idea of the structure and function of the new RUS control structure. It has been very effective in reducing the branching and increasing the efficiency of the RUS parser. We expect to continue to explore the options that this general control structure makes available. A more complete report will be forthcoming.

6. TEMPORAL REASONING

Marc Vilain

The following is a somewhat revised version of a paper [Vilain, 1982] given at the 1982 National Conference on Artificial Intelligence (AAAI 82).

6.1 By Means of Introduction

Imagine a world in which I were not a computer scientist, but an inveterate explorer, a world traveler. In this world I would have been many places, spending at one time several years in Africa before going on to explore the Peruvian Andes. Imagine also that for some time during my African stay, I contracted a case of beriberi. Human beings will naturally deduce that my being ill with beriberi came before my being in Peru. This deduction is typical of the kind of reasoning about time that we have tried to capture in a computer system in current development at BBN. The user of our system makes assertions about the interrelations of events in time. The system in turn deduces new information about the events' interrelations, and makes this information available to the user's queries.

In this paper we will describe the salient features of our system. In particular, we will look at the main representation scheme we have chosen for time (we view time primarily in terms of intervals), and will show how deductions about time can be automated with this representation. The system is built around a truth maintenance mechanism, and we will briefly describe the advantages given by this kind of architecture. Finally, we will

describe how our deduction mechanisms can be gracefully extended to deal with time points and absolute dates.

6.2 A Logic of Time

There are several ways in which human beings understand time (for example as points, intervals, or with respect to calendar dates). In our system, we have chosen to represent time primarily -- though not exclusively -- in terms of intervals. In so doing we have followed the suggestions of James Allen [1981a] that intervals are the most computationally natural way of representing time. Relations between time intervals are described in our system by "operators" in a logic. This logic is an extension of that given in [Allen, 1981a]; at its core it is composed of 13 relational primitives and a large body of inference rules. The primitives describe unambiguously each of the possible ways that two intervals can be related (they can be equal, overlap, one can precede the other, and so forth). The precise meaning of these primitives is most intuitively communicated by a drawing, so we will give their definitions here in a graphic form (see Figure 1).

The relational primitives can be joined into relational vectors; a relational vector describes a disjunctive relation between two time intervals. For example:

A (DURING BEGINS OVERLAPS) B

asserts that interval A is either strictly contained in B (DURING), is contained in B but co-starting with it (BEGINS), or overlaps the "left edge" of B (OVERLAPS). See Figure 2. The

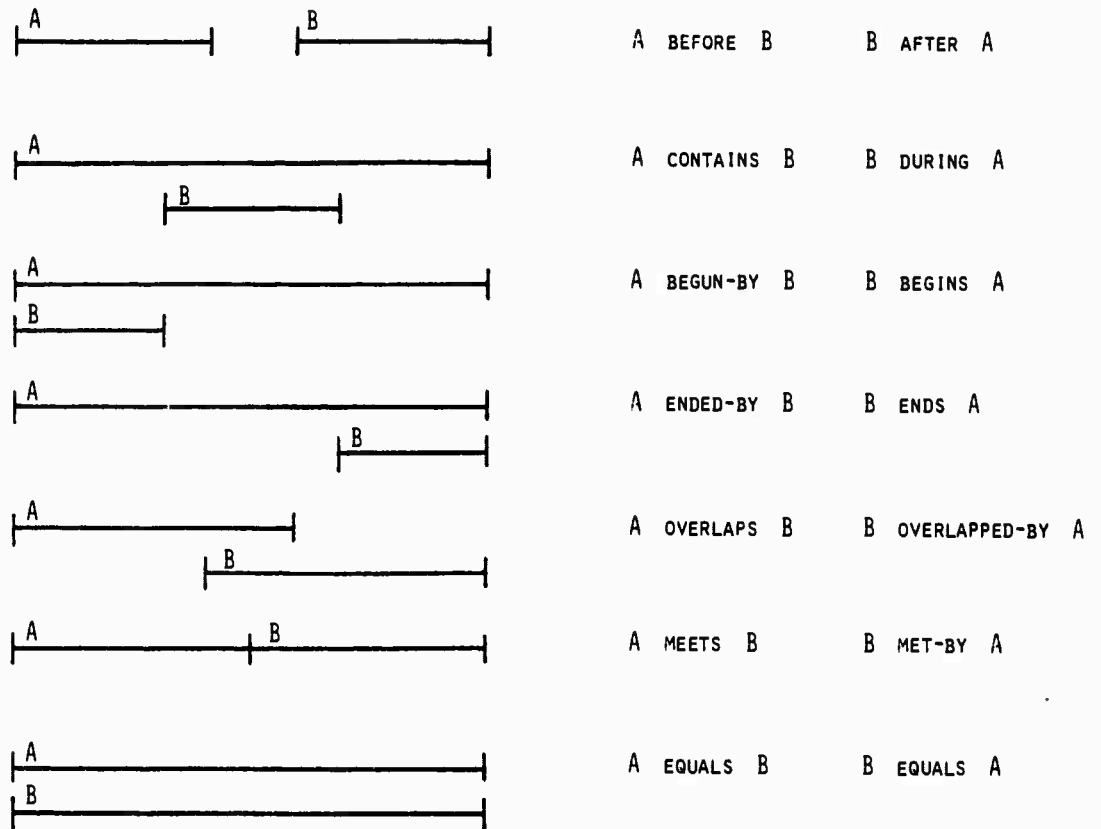


FIG. 1. PRIMITIVE RELATIONS BETWEEN INTERVALS

semantics of relational vectors is one of exclusive disjunction. That is, exactly one and only one of the primitive components of the vector precisely describes the relation of the intervals linked by the vector. Hence, a vector consisting of only one primitive exactly describes the relation between two intervals, whereas the vector composed of all 13 primitives we interpret as the zero-vector. Asserting that two intervals are related by the

zero-vector means that one in fact knows nothing about how they actually relate.

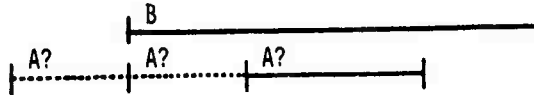


FIG. 2. THE RELATION A (DURING BEGINS OVERLAPS) B

We mentioned above that our logic has as part of its core a body of inference rules. These rules are used to combine known assertions and deduce new information. They have the following form:

"If interval A is related to interval B by R1
and B is related to interval C by R2
then A is related to C by R3" (1)

R1 and R2 are relational primitives and R3 is a vector. The following three rules (illustrated by Figure 3) are typical examples.

A CONTAINS B and B CONTAINS C (2)
=> A (CONTAINS) C

A CONTAINS B and B BEGUN-BY C (3)
=> A (CONTAINS) C

A CONTAINS B and B OVERLAPPED-BY C (4)
=> A (CONTAINS BEGUN-BY OVERLAPPED-BY) C

In our system the rules are used to define the composition properties of the primitive relations of the logic; there is thus one composition rule for each pair of primitive relations (169

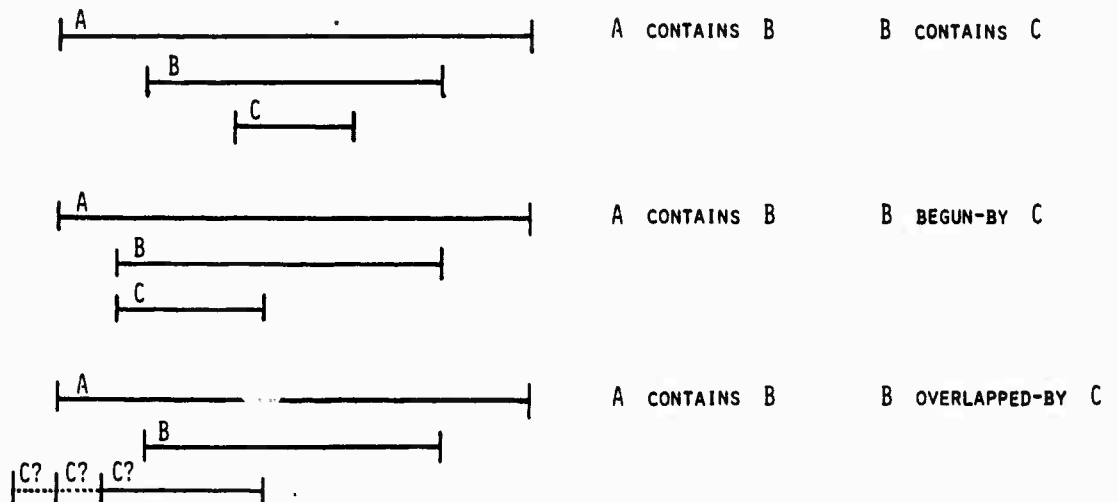


FIG. 3. ILLUSTRATIONS OF RULES 2-4

rules in total). The rules can be extended in a straightforward way to deal with cases where intervals are related by vectors constructed of more than one primitive relation. Consider formula 1 above. Say R_1 is actually a vector $V = (v_1 \dots v_m)$ and R_2 is the vector $U = (u_1 \dots u_n)$. Then R_3 is computed by combining (disjunctively) the vectors deduced from the composition rules for the pairs of primitives v_i and u_j (for each component v_i of V and each component u_j of U). This process preserves the disjunctive semantics of vectors.

For example, say A is related to B, and B is related to C as in these two assertions:

A (CONTAINS) B
B (CONTAINS BEGUN-BY OVERLAPPED-BY) C.

To compute A's relation to C, we combine the deductions made by the three rules above, and obtain the following result.

A (CONTAINS BEGUN-BY OVERLAPPED-BY) C

6.3 Using the Logic

Our system endeavors to maintain a "complete picture" of all the interrelations of all the time intervals the user has declared to exist. That is, for each pair of intervals declared by the user, the system will keep track of the vector that most accurately describes their interrelation. Some of these relation vectors will have been asserted by the user, others must be deduced from the user's original assertions. These deductions are performed in a process of constraint propagation which is guided by the basic composition rules of the time logic. As we saw above, if we know that A relates to B by R1, and B to C by R2, then we can constrain A's relation to C by the composition rule for R1 and R2. If C is also known to relate to D by R3, then we can constrain A's relation to D by composing R3 with the composition of R1 and R2, and so forth. This is illustrated in Figure 4.

As reported elsewhere [Vilain, 1982], the original algorithm we chose to implement the constraint propagation was a variant of one cited in [Aho et al., 1974]. This solution proved to be inadequate (it failed to be complete in certain obscure cases), and we have since replaced it with a version of the well-known graph labeling algorithm of David Waltz [Waltz, 1975]. The Waltz algorithm lends itself well to formal analyses of correctness and efficiency, and we have shown it to have several valuable properties when applied to our representation.

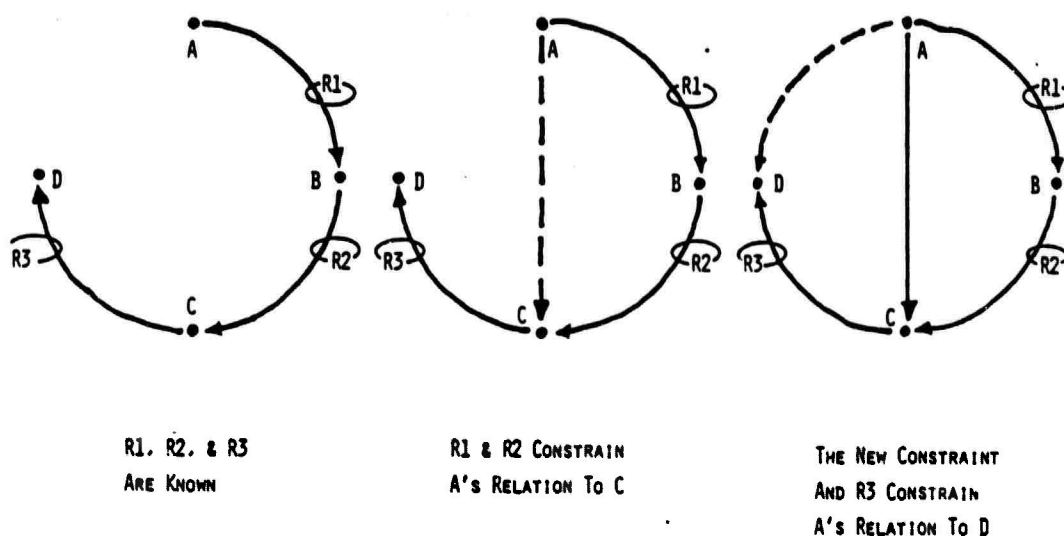


FIG. 4. CONSTRAINT PROPAGATION

The first of these is a completeness result. We have shown that the algorithm has (at least) a limited form of completeness. Namely, any constraint that can be deduced on the basis of the composition rules is in fact deduced by the algorithm. In other words, the program computes the full consequences of the user's assertions as implied by the rules. This completeness result is limited only in that it is not a semantic proof of the completeness of the representation itself. We have yet to show formally that our scheme completely captures the semantics of temporal intervals. This is a hard proof, and is not likely to be immediately forthcoming. However, our experience so far with using the representation has been very positive; we are fully satisfied with its known capacities.

Our version of the Waltz algorithm also has some soundly-

established efficiency characteristics. The relations between n time intervals can be totally constrained by the program in $o(n^3)$ time while using $o(n^2)$ space. This is a firm upper bound, and holds no matter how many assertions the user actually makes. As a rough measure, adding a new time interval to the existing set elicits quadratic behavior from the algorithm.

6.4 Truth Maintenance

One of the basic design criteria in building our time system has been to make it easily usable: we wanted to provide rich facilities for controlling the system and interacting with it. To this extent, we chose to implement it as a truth maintenance system (TMS). Truth maintenance systems have traditionally been used to perform certain forms of propositional deduction [Doyle, 1978, McAllester, 1980]. They have demonstrated many favorable characteristics for that kind of reasoning.

We have adapted the TMS model to the temporal domain, and retained the advantages of the propositional systems. Briefly, they are as follows:

Incremental description refinement. The system updates its temporal knowledge base incrementally as the user adds or removes assertions. This allows the user to build up a description of the interrelations of intervals in bits and pieces, and refine it over time. The user may also assert hypothetical relations and retract them at a later moment.

Deduction justification. Whenever the system performs a deduction (either on the basis of an assertion or as a

consequence of another deduction), it records a justification for the new result. Thus any conclusion reached by the system is explicitly supported by a set of justifications. These justifications are in turn ultimately supported by an explicit set of user assertions. It is possible for the system to explain why it performed a given deduction by following the deduction's support structure and returning the original set of assertions that underlie the result.

Efficient retraction. By explicitly remembering justifications for its deductions, the system can perform very efficient retractions. To retract the consequences of a user assertion the system simply follows the chain of justifications supported by the assertion, cancelling any deductions present in the chain. Other authors have referred to this kind of retraction as dependency-directed backtracking [Doyle, 1978, McAllester, 1980].

Inconsistency resolution. Our system builds up its temporal knowledge base incrementally, on the basis of a sequence of assertions by the user. In theory, nothing prevents the user from trying to add an assertion that contradicts earlier assertions or consequences that the system has derived from earlier user statements. This situation, however, can easily be detected by the system before the assertion is actually added. The system can then help the user resolve the inconsistency it detected by returning the set of earlier statements that the new assertion contradicts. The user can then choose to retract one or more of these earlier assertions and thereby make the new one no longer inconsistent.

6.5 Time Points

At the onset of this paper we noted that intervals are not the only mechanism by which human beings understand time; another common construct is that of time points. Time points are naturally defined by the boundaries of intervals and by certain dating schemes (which we describe below). In fact, much of the earlier literature on reasoning about time describes computer systems whose primary representation of time was in terms of points, not intervals. This is the case with the CHRONOS system of Bruce [1972] and the time specialist of Kahn and Gorry [1977].

Our system handles time points in much the same way that it handles intervals: points are objects whose interrelations can be described by primitives in a logic. The logic of points is arrived at by expanding the earlier logic of intervals. To the older logic we add new primitive relations (which like the old ones can be built into vectors), and new composition rules over these primitives (which can be "conjoined" to deal with vectors). The new primitives can be broken into three groups:

1. Those which relate points to other points,
2. Those which relate intervals to points, and
3. Those which relate points to intervals.

As before, we prefer to define these new relations graphically (see Figure 5).

The composition rules that we add to the logic not only define the composition of the new primitive relations with themselves, but also with the original relations that applied to intervals only. Again, we present some typical examples of these rules (illustrated by Figure 6).

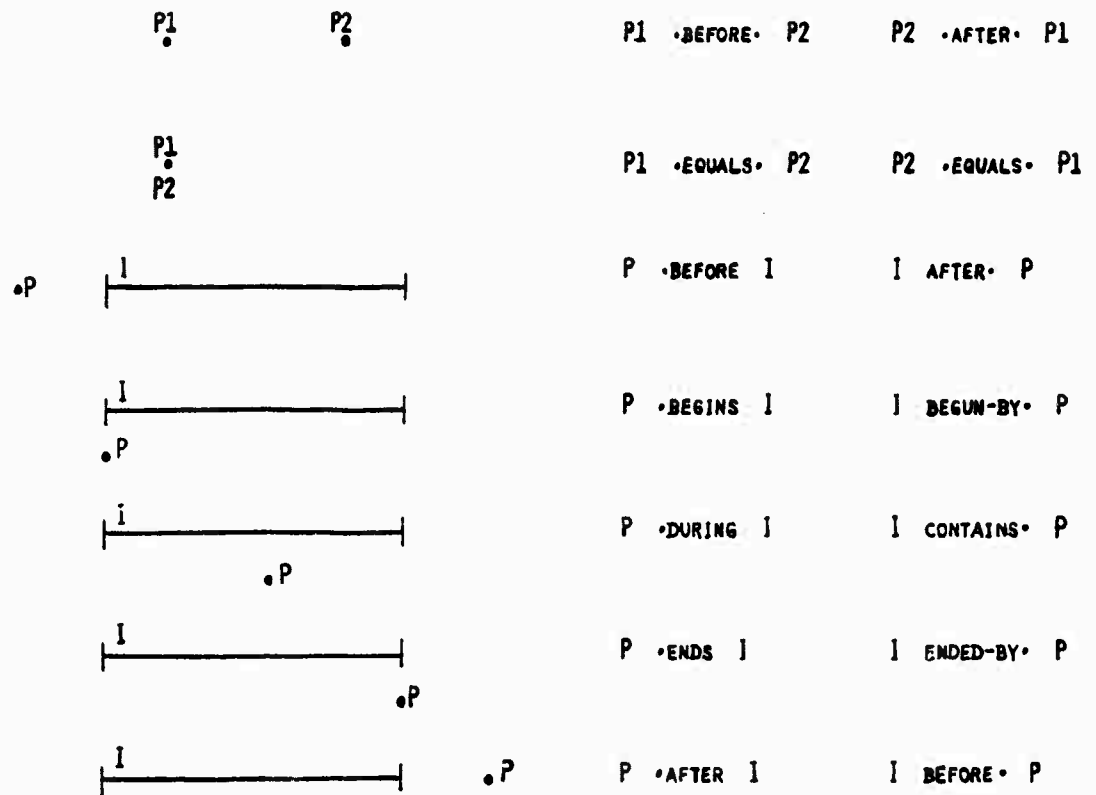


FIG. 5. NEW PRIMITIVE RELATIONS (INVOLVING POINTS)

A BEFORE• P1 and P1 •BEFORE• P2 (5)
=> A (BEFORE•) P2

A BEFORE• P and P •BEFORE B (6)
=> A (BEFORE) B

A BEGINS B and B CONTAINS• P (7)
=> A (CONTAINS• ENDED-BY• BEFORE•) P

The mechanism by which our system makes deductions about

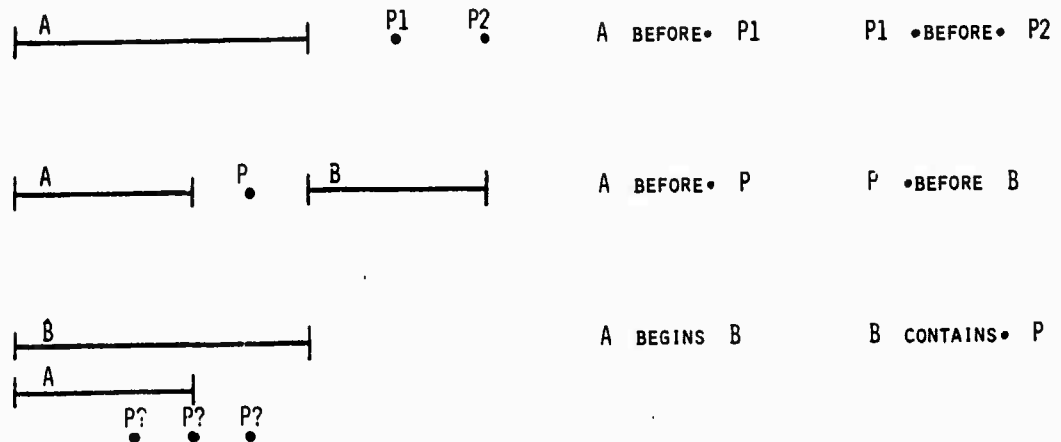


FIG. 6. ILLUSTRATIONS OF RULES 5-7

points is just an extension of that which it uses to make deductions about intervals. As with intervals, the user can declare the existence of certain time points and assert their interrelations to other points or to intervals. Just as before, the system maintains a "complete picture" of all these objects' interrelations by means of a constraint propagation operation. The operation is simply performed using the expanded set of composition rules in the newer logic.

As a final note about points, we should state that including them along with intervals in the domain of our system only minimally complicates the deduction algorithms. The truth maintenance mechanisms remain unaffected, as do the complexity and completeness results.

6.6 Absolute Dating

There are two dating mechanisms that are commonly used by people. The first dates entire intervals, and its best example is the standard calendar (which gives a unique name to intervals of an entire day). The second assigns "time stamps" of sorts to particular moments or points in time. This kind of dating is exemplified by the reference to "9:00 o'clock" in the sentence "Bill will arrive by 9:00 o'clock". The time stamps assigned by this method of dating are what we call absolute dates.

Our system incorporates a method for reasoning about absolute dates. Our system handles statements about absolute dates by mapping them into the logic of intervals and points. Once this mapping is completed, the original statements involving absolute dates need in fact never be consulted again.

More specifically, whenever the user makes an assertion relating an interval (or point) to a date, the system automatically generates a time point to correspond to the date. This generated time point (which we call a date point) is then appropriately related to the interval (or point) in the user's assertion. The new date point must also be related to all other known date points; the system performs this automatically by simply adding a few new statements to its store of assertions. This process is performed under the guidance of a simple calendar function.

Once the system has generated these (internal) assertions, it can use them to deduce new information by the very same constraint propagation process that operates over intervals and points. It never again need consult the user's original

statements relating dates to intervals or points. This is an appealing result since it obviates the need to maintain separate mechanisms for dealing with dated and undated information. This dual reasoning was typically present in earlier time systems, such as that of Kahn and Gorry [op. cit.].

Finally, we should note that the assertions our system generates when creating a date point have the same computational status as the user's undated assertions. This insures that dated assertions will be subject to the TMS justification and inconsistency detection mechanisms, just as undated ones are.

6.7 Other Aspects of the System

There are a number of other features of our time representation system which we will only mention here in passing. Most of these are concerned with increasing the system's overall efficiency and ease of use. We have implemented a number of mechanisms for limiting the total amount of computation performed by the system. We have also introduced some specialized constructions that allow the user to cluster or package information in useful and space-saving ways. These features are not yet completely established. Any decisions on their final form must wait until we have tested them in practice. They will be reported on more fully in a forthcoming document [Vilain, forthcoming].

In closing, we would like to place our work in a broader perspective. Recently, several writers have described general models of time and action (specifically James Allen [1981b] and Drew McDermott [1981]). Our efforts are nowhere nearly as

ambitious as theirs. Instead we have sought to construct a basic computational tool that could be used by larger programs.² Our approach is actually consistent with that of McDermott and that of Allen. In fact, both of these authors have assumed in their models the existence of underlying time maintenance modules similar to the one described here.

Our goals in this research have all along been to provide a simple but complete inference mechanism over the time domain, one that we hoped would free researchers in AI from having to tackle the low-level details of reasoning about time. We are hoping that our system will permit them to turn their attention to more rewarding investigations in problem solving, language understanding, and other intelligent behavior.

²In the BBN natural language project, we intend to use our time system as part of the plan recognizer. We also expect it to be a crucial building block of the response planner we are currently developing.

7. PUBLICATIONS

Bates, M. and Sidner, C.L., "A Case Study of a Method for Determining the Necessary Characteristics of a Natural Language Interface," (Proceedings of the ECICS-82), in Sandewall (ed.), Integrated Interactive Systems, North Holland, forthcoming.

Abstract

There is no generally accepted (or even commonly used) method of determining, in the early stages of a design process, just what capacities a particular natural language interface must possess in order to be effective. In this paper we set forth a case study of a methodology that has been extremely effective in our (graphically oriented) domain and which can easily be adapted to other situations.

This paper presents three aspects of our research: (1) a description of the methodology that we used in developing and analyzing an extended prototypical dialogue between a user and such a system, (2) portions of our analysis of that dialogue that present both the information obtained and the method of obtaining it, and (3) conclusions about the necessary linguistic and non-linguistic capacities of an intelligent conversational partner, as drawn from the full analysis.

Bobrow, R. and Bates, M., "Design Dimensions for Non-Normative Understanding Systems," position paper in Proceedings of the 20th Annual Meeting of the Association of Computational Linguistics, Toronto, Canada, June 16-18, 1982.

Abstract

We pose a number of questions in order to clarify the theoretical and practical issues involved in building "non-normative" natural language systems. We give brief indications of the range of plausible answers, in order to characterize the space of decisions that must be made in designing such a system. The first questions cover what is intended

by the ill-defined term "non-normative system", beyond the important but vague desire for a "friendly and flexible" computer system. The remaining questions cover several of the architectural issues involved in building such a system, including the categories of knowledge to be represented in the system, the static modularization of these knowledge sources, and the dynamic information and control flow among these modules.

Israel, D.J. and Brachman, R.J., "Distinctions and Confusions: A Catalogue Raisonne," Proceedings of the 7th International Joint Conference on Artificial Intelligence, August 1981.

Abstract

It's been said many times that semantic nets are mere notational variants of "predicate calculus." But before we lay down our nets, and embrace Logic, we ought at least to be clear about what "predicate calculus" is. We advance some clarificatory points to this effect. Moreover, there seems to be at least one feature of some net/frame schemes that defy simple translation into standard formalisms: "prototypes". Yet this feature is not itself without difficulties, some of which we address here. In the end, we plead for an open mind and a search for formal-semantical alternatives to "classical logic".

Schmolze, J.G., and Brachman, R. J., "Proceedings of the 1981 KL-ONE Workshop," Report No. 4842, Bolt Beranek and Newman Inc., June 1982.

Abstract

The second KL-ONE Workshop gathered researchers from twenty-one universities and research institutions for a series of discussions and presentations about the KL-ONE knowledge representation language. These proceedings summarize the discussions and presentations, provide position papers from the participants, list the agendas of the Workshop along with the names and addresses of the participants, and include a description of the KL-ONE language plus an index of some KL-ONE technical terms.

Sidner, C.L., "Focusing for the Interpretation of Pronouns," American Journal of Computational Linguistics, 7:4(217-231), 1982.

Abstract

Recent studies in both artificial intelligence and linguistics have demonstrated the need for a theory of the comprehension of anaphoric expressions, a theory that accounts for the role of syntactic and semantic effects, as well as inferential knowledge in explaining how anaphors are understood. In this paper a new approach, based on a theory of the process of focusing on parts of the discourse, is used to explain the interpretation of anaphors. The concept of a speaker's foci is defined, and their use is demonstrated in choosing the interpretations of personal pronouns. The rules for choosing interpretations are stated within a framework that shows: how to control search in inferring by a new method called constraint checking; how to take advantage of syntactic, semantic and discourse constraints on interpretation; and how to generalize the treatment of personal pronouns, to serve as a framework for the theory of interpretation for all anaphors.

Sidner, C.L., "Focusing in the Comprehension of Definite Anaphora," in B. Berwick and M. Brady (eds.) Computational Models of Discourse, MIT Press, 1983.

Sidner, C.L., "What the Speaker Means: The Recognition of Speaker Plans in Discourse," to appear in Journal of Mathematics and Computers, 1983.

Abstract

Human conversational participants depend upon the ability of their partners to recognize their intentions, so that those partners may respond appropriately. In such interactions, the speaker encodes his intentions about the hearer's response in a variety of sentence types. Instead of telling the hearer what to do, the speaker may just state his goals, and expect a response that meets these goals at least part way. This paper presents a new model for

recognizing the speaker's intended meaning in determining a response. It shows that this recognition makes use of the speaker's plan, his beliefs about the domain and about the hearer's relevant capacities.

Sidner, C.L. and Bates, M., "Requirements for Natural Language Understanding in a System with Graphic Displays," Report No. 5242, Bolt Beranek and Newman Inc., forthcoming. This work is summarized in Section 4.1 of this report.

Sidner, C.L., "Protocols of Users Manipulating Visually Presented Information with Natural Language," Report No. 5128, Bolt Beranek and Newman Inc., September 1982.

Abstract

This document contains a complete set of transcripts for a set of protocols collected at BBN by Candy Sidner, with the help of Rusty Bobrow and Jeff Gibbons, in the spring of 1980. In all, eight protocols were collected, two preliminary ones and six main ones. The two preliminary protocols are based on the task of designing a 1-bit adder and a 4-bit parallel adder as are three of the main protocols, while the other three make use of KL-ONE as a database system with graphic representation. The purpose of these protocols, and hence the design of the tasks, was to obtain data about how people talk about graphically presented material which they are trying to manipulate in some way. In particular, we were interested in the kinds of references people made and what sorts of instructions they gave to the machine.

Sidner, C.L., "Focusing and Discourse," Discourse Processes, forthcoming (final draft completed for publication).

Abstract

In a discourse, speakers center their attention on a particular element of the discourse, and they talk about it over one or more sentences of the discourse. This element is called the focus, and the process by which speakers center is focusing.

Focusing is a cognitive process which is active during the interpretation of discourse rather than during the interpretation of isolated sentences. To help the hearer determine how successive sentences are related, the speaker uses anaphora to signal the same focus rather than re-introducing in each sentence a noun phrase describing the element of discourse under discussion. This paper describes a process model of focusing that specifies that syntactic, semantic and world knowledge constraints are needed for the hearer to track the speaker's focus in a discourse. The paper illustrates that focusing is a well constrained behavior for speakers, and argues that focusing is a necessary condition for maintaining Grice's maxim of conversation.

Vilain, M.B., "A System for Reasoning about Time," in Proceedings of the 2nd National Conference on Artificial Intelligence (AAAI-82), August, 1982, pp. 197-201. (An updated version of this paper is included in Section 4.2 of this report.)

Presentations

Sidner, C.L., "The Pragmatics of Non-Anaphoric Noun Phrases," unpublished paper presented at Joint NSF-CNRS Seminar on Discourse Processes, June 28-July 2, 1982, Cadarache, France.

Other Relevant Papers

Israel, D.J., "On Interpreting Semantic Network Formalisms," to appear in a special issue on Computational Linguistics (N. Cercone, ed.) of the International Journal of Computers and Mathematics (early 1983). Also Report No. 5117, Bolt Beranek and Newman Inc., September 1982.

Abstract

An attempt to sketch adequate semantic accounts for at least two (kinds of) semantic network formalisms: one, based on the notion of inheritance; one, not. A crucial condition of adequacy to be satisfied is fidelity to some of the intuitions of the creators of the formalisms.

Israel, D.J., "Taxonomies of Natural Kinds," appearing as a technical addendum to BBN Report No. 5117 above.

Abstract

A serious look at the structure of taxonomies all of whose nodes are natural kinds, motivated by work of biologists and anthropologists, and by "folk taxonomies" of real, live folk. Multiple inheritance leads to graph structures which are upper semilattices. These semilattices can be united by introducing a TOP (highest genus) and the semilattices converted to a lattice by introducing a BOTTOM. The resulting lattices are neither distributive, modular, nor complemented, and the meet and join operations of taxonomic lattices are different than conjunction, disjunction, intersection and union.

Israel, D.J., "Some Remarks on the Semantics of Representation Languages" (with R.J. Brachman), to appear as a chapter in the collection "Perspectives on Conceptual Modelling," (M. Brodie, J. Mylopoulos, and J. Schmidt, eds.), to be published by Springer Verlag.

Abstract

It has been said many times that semantic nets are mere notational variants of predicate calculus. But before we lay down our nets, we ought at least to be clear about what predicate calculus is. We will attempt to make some clarifications in this regard. We also devote some attention to the notions of semantic nets. In the end, we simply plead for an open mind.

Israel, D.J., The Preface for Computational Models of Discourse, (M. Brady and B. Berwick, eds.), volume in MIT AI series (to be published in December, 1982).

Sidner, C.L. and Vittal, J., "Knowledge Representation Tools for Design, Training and Use of Information Systems," (Proceedings of the ECICS-82), in Sandewall (ed.), Integrated Interactive Systems, North Holland, forthcoming.

Abstract

Information systems are generally used for decision making tasks, but users of those systems are severely limited in their interactions with them. We are intending to provide an aid to creating and using an information system. For such an aid to succeed, users and designers must have a feedback channel with a common language in which to communicate. A prerequisite of the language is a model of the system that supplies a tool for new users for understanding the system and which is described from the user's view of his task. This model can then also be used to help train people in the use of the system, or to help them understand the differences between two versions of the same system.

One way to create such a system is through the use of a knowledge representation language. This language will provide a mechanism for the representation and implementation of models (e.g., a user's model of a task and a system designer's model of the system). Also central to the system are processes to manipulate these representations and build additional ones to note the differences between the models and to train users in using the system.

Woods, W.A., "Optimal Search Strategies for Speech Understanding Control," in Artificial Intelligence, Vol. 18, No. 3, May 1982. (Also in B. Webber and N. Nilsson (eds.), Readings in Artificial Intelligence, Tioga Publishing Co., Palo Alto, CA, 1981.)

Abstract

This paper describes two algorithms for finding the optimal interpretation of an unknown utterance in a continuous speech understanding system. These methods guarantee that the first complete interpretation found will be the best scoring interpretation possible. Moreover, unlike other optimal strategies, they do not make finite-state assumptions about the nature of the grammar for the language being recognized. One of the methods, the density method, is especially interesting because it is not an instance of the 'optimal' A* algorithm of Hart, Nilsson, and Raphael, and appears to be

superior to it in the domains in which it is applicable. The other method, the shortfall method, is an instance of the A* algorithm using a particular heuristic function. Proofs of the guaranteed discovery of the best interpretation and some empirical comparisons of the methods are given. The relationship of these methods to strategies used in existing speech understanding systems is also discussed. Although presented in the speech context, the algorithms are applicable to a general class of optimization and heuristic search problems.

Woods, W.A., "Transition Network Grammars for Natural Language Analysis" (Communication of the ACM, October 1970, pp. 591-606), in Yoh-Han Pao and George W. Ernest (eds.), Tutorial: Context-Directed Pattern Recognition and Machine Intelligence Techniques for Information Processing, IEEE Computer Society Press, Silver Springs, MD, 1982.

Woods, W.A., "HWIM: A Speech Understanding System on a Computer," in Michael A. Arbib, D. Caplan, and J. Marshall (eds.), Neural Models of Language Processes. New York: Academic Press, 1982.

Abstract

This chapter is written in the belief that the space of hypotheses through which the computer searches serially in seeking to provide an interpretation for an utterance is similar to that through which the brain searches, presumably in parallel. We motivate this claim by looking at a number of experiments on human speech understanding, and the way in which these led to the design of a computerized speech understanding system called HWIM ("Hear What I Mean"). The system organization, control strategy, grammar, and network representations of HWIM are explained, and the system operation is illustrated with an analysis of the utterance "Do we have a surplus?".

8. BIBLIOGRAPHY AND REFERENCES

- Aho, A.V., Hopcroft, J.E. and Ullman, J.D. [1974]. The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading, MA.
- Alexander, D. and Kunz, W.J. [1964a]. Some Classes of Verbs in English. Linguistics Research Project, Indiana University, F.W. Householder, Jr., Principal Investigator, distributed by Indiana University Linguistics club, Bloomington.
- Alexander, D. and Matthews, P.H. [1964b]. Adjectives before that-Clauses in English. Linguistics Research Project, Indiana University, F.W. Householder, Jr., Principal Investigator, distributed by Indiana University Linguistics Club, Bloomington.
- Allen, J.F. [1981a]. Maintaining Knowledge about Temporal Intervals. Technical Report 86, University of Rochester, Dept. of Computer Science, January.
- Allen, J.F. [1981b]. A General Model of Action and Time. Technical Report No. 97, University of Rochester, Dept. of Computer Science, September.
- Bates, M., Bobrow, R.J. and Webber, B.L. [1981]. Tools for Syntactic and Semantic Interpretation. BBN Report No. 4785, Bolt Beranek and Newman Inc.
- Bates, M. and Bobrow, R.J. [forthcoming]. The RUS Parsing System. Bolt Beranek and Newman Inc.
- Bates, M. and Sidner, C.L. [1982]. A Case Study of a Method for Determining the Necessary Characteristics of a Natural Language Interface. Proceedings of the ECICS-82, in Sandewall (ed.), Integrated Interactive Systems, North Holland, forthcoming.
- Bobrow, R.J. [1978]. The RUS System. BBN Report No. 3878, Bolt Beranek and Newman Inc.
- Bobrow, R.J. and Webber, B.L. [1980]. PSI-KLONE - Parsing and Semantic Interpretation in the BBN Natural Language Understanding System. CSCSI/CSEIO Annual Conference, CSCSI/CSEIO.

- Bobrow, R.J. and Webber, B.L. [1980]. Knowledge Representation for Syntactic/Semantic Processing. Proceedings of the First Annual National Conference on Artificial Intelligence, American Association for Artificial Intelligence.
- Brachman, R.J. [1978]. A Structural Paradigm for Representing Knowledge, BBN Report No. 3605, Bolt Beranek and Newman Inc., May.
- Brachman, R.J. [1979a]. On the Epistemological Status of Semantic Networks. In Findler, N.V. (ed.), Associative Networks: Representation and Use of Knowledge by Computers, pp. 3-50. Academic Press, New York.
- Brachman, R.J. [1979b]. Encoding Knowledge in Partitioned Networks. In Findler, N.V. (ed.), Associative Networks: Representation and Use of Knowledge by Computers, pp. 51-92. Academic Press, New York.
- Brachman, R.J. [1980]. "I lied about the trees." Unpublished manuscript.
- Bridgeman, L.I., Dillinger, D., Higgins, C., Seaman, P.D., Shank, F.A. [1965a]. More Classes of Verbs. Linguistics Research Project, Indiana University, F.W. Householder, Jr., Principal Investigator, distributed by Indiana University Linguistics Club, Bloomington.
- Bridgeman, L.I., Dillinger, D., Higgins, C., Seaman, P.D., Shank, F.A. [1965b]. Further Classes of Adjectives. Linguistics Research Project, Indiana University, F.W. Householder, Jr., Principal Investigator, distributed by Indiana University Linguistics Club, Bloomington.
- Bruce, B.C. [1972]. A Model for Temporal References and Its Application in a Question Answering Program. Artificial Intelligence 3(1): 1-25.
- Chomsky, A.N. [1975]. The Logical Structure of Linguistic Theory. Microfilmed, M.I.T. Libraries, partially published by Plenum Press, New York.
- Doyle, J. [1978]. Truth Maintenance Systems for Problem Solving. Technical Report No. 419, Artificial Intelligence Laboratory, M.I.T., January.
- Fahlman, S.E. [1979]. NETL: A System for Representing and Using Real-World Knowledge.

- Fromkin, V.A. [1973]. Janua Linguarum, Series major. Vol. 77: Speech Errors as Linguistic Evidence. Mouton, The Hague.
- Goodwin, J.W. [1979]. Taxonomic Programming with Klone. Technical Report LiTH-MAT-R-79-5, Informatics Laboratory, Linköping University, Linköping, Sweden.
- Harris, Z. [1957]. "Co-occurrence and Transformation in Linguistic Structure. Language 33, pp. 283-340.
- Kahn, K. and Gorry, G. A. [1977]. Mechanizing Temporal Knowledge. In Artificial Intelligence 9(1):87-108.
- Marcus, M. [1980]. A Theory of Syntactic Recognition for Natural Language M.I.T. Press.
- McAllester, D.A. [1980]. An Outlook on Truth Maintenance. A.I. Memo 551, M.I.T., Artificial Intelligence Laboratory, February.
- McDermott, D. [1981]. A Temporal Logic for Reasoning about Processes and Plans. Research Report 196, Dept. of Computer Science, Yale University, March.
- O'Malley, M.H. [1975]. The Children of Lunar: an exercise in comparative grammar. Memorandum No. ERL-543, SUR Note No. 189, Computational Speech and Language Group, Electronics Research Laboratory, University of California at Berkeley.
- Proctor, P. et al., eds. [1978]. Longman Dictionary of Contemporary English, Longman Group Limited, Harlow and London.
- Quirk, R. and Greenbaum, S. [1973]. A Concise Grammar of Contemporary English Harcourt Brace Jovanovich, Inc., New York.
- Quine, W.V.O. [1960]. Word and Object. The M.I.T. Press, Cambridge, MA.
- Reiter, R. [1980]. A Logic for Default Reasoning. Artificial Intelligence 13 (1,2): 81-132, April.
- Sager, N. [1981]. Natural Language Information Processing: A Computer Grammar of English and Its Applications. Addison-Wesley Publishing Co., Inc., Reading, MA.

- Schmolze, J.G., and Brachman, R.J. [1982]. Proceedings of the 1981 KL-ONE Workshop. Report No. 4842, Bolt Beranek and Newman Inc., June.
- Schubert, L.K., Goebel, R.G., and Cercone, N.J. [1979]. The Structure and Organization of a Semantic Net for Comprehension and Inference. In Findler, N.V. (ed.), Associative Networks: Representation and Use of Knowledge by Computers, pp. 121-175. Academic Press, New York.
- Sellars, R.W. [1917]. The Essentials of Logic. The Riverside Press, Cambridge, MA.
- Sidner, C.L. [1979]. Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse. Technical Report AI-TR-537, Artificial Intelligence Laboratory, M.I.T., June.
- Smith, B.C. [1978]. Levels, Layers, and Planes: The Framework of a Theory of Knowledge Representation Semantics. Master's thesis, Artificial Intelligence Laboratory, M.I.T., February.
- Smith, B.C. [1982]. Reflections and Semantics in a Procedural Language. Ph.D. thesis, Artificial Intelligence Laboratory, M.I.T., January.
- Sondheimer, N.K. and Weischedel, R.M. [1980]. A Rule-Based Approach to Ill-Formed Input. Proceedings of 8th International Conference on Computational Linguistics, pp. 46-54, Tokyo, Japan, October.
- Vilain, M.B. [1982]. A System for Reasoning about Time. In Proceedings of the National Conference on Artificial Intelligence (AAAI82), pp. 197-201. The American Association for Artificial Intelligence.
- Vilain, M.B. [forthcoming]. A Truth Maintenance System for Time. Technical Report, Bolt Beranek and Newman Inc.
- Waltz, D.L. [1975]. Understanding Line Drawings of Scenes with Shadows. In Winston, P.H. (ed.), The Psychology of Computer Vision, pp. 19-91. McGraw Hill, New York.
- Weischedel, R.M. and Black, J.E. [1980]. If the Parser Fails. Proceedings of the 18th Annual Meeting of the ACL, June.

Weischedel, R.M. and Sondheimer, N. [1981]. A Framework for Processing Ill-Formed Input. Dept. of Computer and Information Sciences, University of Delaware, October.

Woods, W.A. [1975]. What's in a Link: Foundations for Semantic Networks. In Bobrow, D.G. and Collins, A. (eds.), Representation and Understanding: Studies in Cognitive Science, pp. 35-82. Academic Press, New York.

Woods, W.A. [1980]. Cascaded ATN Grammars, In American Journal of Computational Linguistics 6, 1, January-March.

Woods, W.A. [1982]. Optimal Search Strategies for Speech Understanding Control. In Artificial Intelligence 18, 3, June.

Official Distribution List

Contract N00014-77-C-0378

	<u>Copies</u>
Defense Documentation Center Cameron Station Alexandria, VA 22314	12
Office of Naval Research Information Systems Program Code 437 Arlington, VA 22217	2
Office of Naval Research Code 200 Arlington, VA 22217	1
Office of Naval Research Code 455 Arlington, VA 22217	1
Office of Naval Research Code 458 Arlington, VA 22217	1
Office of Naval Research Branch Office, Boston 495 Summer Street Boston, MA 02210	1
Office of Naval Research Branch Office, Chicago 536 South Clark Street Chicago, IL 60605	1
Office of Naval Research Branch Office, Pasadena 1030 East Green Street Pasadena, CA 91106	1
Naval Research Laboratory Technical Information Division Code 2627 Washington, D.C. 20380	6

cont'd.

Naval Ocean Systems Center Advanced Software Technology Division Code 5200 San Diego, CA 92152	1
Dr. A. L. Slafkosky Scientific Advisor Commandant of the Marine Corps (Code RD-1) Washington, D.C. 20380	1
Mr. E. H. Gleissner Naval Ship Research & Development Ctr. Computation & Mathematics Dept. Bethesda, MD 20084	1
Capt. Grace M. Hopper, USNR Naval Data Automation Command Code OOH Washington Navy Yard Washington, D.C. 20374	1
Mr. Paul M. Robinson, Jr. NAVDAC 33 Washington Navy Yard Washington, D.C. 20374	1
Advanced Research Projects Agency Information Processing Techniques 1400 Wilson Boulevard Arlington, VA 22209	1
Capt. Richard L. Martin, USN 3311 West Avenue Newport News, VA 23607	1
Director National Security Agency Attn: R54, Mr. Page Fort G.G. Meade, MD 20755	1
Director National Security Agency Attn: R54, Mr. Glick Fort G.G. Meade, MD 20755	1

cont'd.

Major James R. Kreer
Chief, Information Sciences
Dept. of the Air Force
Air Force Office of Scientific Research
European Office of Aerospace
Research & Development
Box 14
FPO New York 09510

1

Dr. Martin Epstein
National Library of Medicine
Bldg. 38A, 8th Floor
8600 Rockville Pike
Bethesda, MD 20209

1